Model Predictive Control of Dynamically Substructured Systems with Application to a Servohydraulically-Actuated Mechanical Plant *

Guang Li, David P. Stoten and Jia-Ying Tu †

June 2, 2010

Abstract

Dynamically substructured systems (DSS) are increasingly used by the dynamics testing community. DSS involves the physical testing of full size critical components in parallel with numerical testing of the remaining components. This has certain advantages over other testing methods [24]. However, the synchronization of the signals at the interface between the physical and numerical substructures of DSS requires a high fidelity controller. In practice, the performance of the DSS testing can be significantly degraded by input saturation of the actuators. In this paper, we use model predictive control (MPC) to cope with the saturation problem in DSS. To facilitate the MPC and observer design for DSS, a modified DSS framework based on an existing one is proposed [22]. As a case study, a quasi-motorcycle (QM) system is converted into the modified DSS framework and a traditional on-line MPC control strategy is implemented in real-time.

Key words: *Predictive control; Dynamically substructured systems; Multivariable control.*

1 Introduction

In real-time dynamic testing of a system, some critical components can be either too complicated to be numerically modelled, due to the presence of uncertainties and nonlinearities, or too

*This is the final draft of the paper published by IET Control & Applications vol.4, no.2, pages 253-264, February 2010.

†The authors are with Advanced Control and Test Laboratory (ACTLab), Department of Mechanical Engineering, The University of Bristol, Queens Building, University Walk, Bristol, BS8 1TR, UK. Email: guangli78@gmail.com, D.P.Stoten@bristol.ac.uk, Jiaying.Tu@bristol.ac.uk

difficult to be tested in a laboratory environment due to the cost or size (for example, the testing of large-scale engineering structures such as bridges and dams). To circumvent these problems, the use of the dynamic substructuring concept in real-time experimental testing has become an appealing strategy in recent years [13]. The principal idea of substructuring is to simultaneously test the complex critical components of the system (represented as a physical substructure) in real-time and the remainder of the system as a numerical model (represented as a numerical substructure). This leads to a dynamically substructured system (DSS). The DSS testing can be more advantageous than the existing testing methods such as full-size testing of the entire system, scale-model testing, pseudo-dynamic testing and purely numerical testing, etc. (see [24, 27] and the references therein).

The main feature of DSS testing is that its physical and numerical substructures are tested separately, but not independently. The two substructures are linked to each other by the interaction constraints at their interfaces. Hence an important issue of the substructuring method is the synchronization of the substructures, which significantly affects the testing accuracy of the entire system. This demands a high fidelity of control to reduce the error of the output variables of the substructures at the interface, while satisfying the constraint signals, so that the DSS responds as close as possible to the original emulated system.

Successful existing control strategies for DSS include Linear Substructuring Control (LSC) and the adaptive Minimal Control Synthesis (MCS) algorithm [14, 21–24, 26]. Dynamic interaction between the substructures, together with the dynamics of the transfer system (and its associated actuators), will normally cause problems with synchronization. These problems can be from the dynamic uncertainties, measurement noise and nonlinearities. In addition, one common and important problem is caused by saturation of the actuators in the transfer systems. Actuator saturation is a fundamental problem in control and significant theoretical work and practical investigations of this issue have been presented in the literature (see, e.g. [4], [6], [7] and the references therein). However, the problem of coping with actuator saturation within DSS has received scant attention. One approach to control the multivariable DSS, while explicitly considering the actuator constraints, is an anti-windup compensator developed in [9]. In that work, based on a pre-designed linear controller, e.g. an LQR or H_∞ controller, an anti-windup compensator was synthesized by directly minimizing the L_2 gain from the testing signal to the DSS substructuring error.

In this paper we apply another control strategy to DSS, Model Predictive Control (MPC),

as an alternative approach to dealing with actuator constraints. The main motivation is that MPC can lead to an optimal solution, compared with the anti-windup approach in [9]. MPC is also easier to tune, since it can inherently handle constraints present in multivariable systems. Traditionally, MPC is an on-line control strategy, which solves an optimization problem at each sampling instant, with the current state as the initial state. However, due to the heavy on-line computational burden, the initial applications of MPC were restricted to relatively slow process control problems in chemical industries [15]. One approach to reduce this burden is to replace the on-line optimization with an off-line procedure, by establishing a look-up table from the estimated states to the control inputs [1]. This is achieved by using the piecewise affine properties of the MPC controller. However, the computational complexity and the required memory space can increase rapidly with the growth of the problem size (see, e.g. [29, 30], for a detailed discussion on a comparison of the two control strategies). In recent years, with the development of new, efficient optimization algorithms and the rapid progress of hardware computing ability, a large number of applications of the traditional on-line MPC to fast systems have been reported in areas such as aerospace, power plants and the automotive industry (see [16] for a survey). These promising results motivated the implementation of the on-line MPC to DSS for real-time testing of electro-mechanical components, which normally demand a high sampling rate.

This paper presents the application and implementation of on-line MPC to the DSS testing of a servohydraulically actuated mechanical system. To facilitate the MPC design for DSS, a framework modified from [22–24] is proposed. This modified framework strictly separates the numerical and physical substructures in DSS. One of the benefits of using this modified framework is the ability to design a reduced order observer for DSS, which helps to reduce the on-line computation time. In the case study, a quasi-motorcycle (QM) suspension system developed at the University of Bristol was tested in real-time. The on-line MPC optimization was solved by using the active set algorithm programmed in C, [28]. This code was recently implemented successfully in the application of MPC, with a prediction horizon of 10, to an active structure consisting of an SISO system with 18 states, using sampling rates up to 5 kHz on a 200 MHz DSP [29]. In our implementation, the code was embedded into an S-function in SIMULINK®, which was compiled and implemented by a dSPACE©real-time control system. In this case, the DSS of the QM suspension system has 12 states, 2 inputs and 2 outputs, while the MPC controller can be implemented with a prediction horizon of 5 at a sampling

rate up to 1.2 kHz (depending on different formulations of MPC controller). The experimental results demonstrated the advantage of applying MPC over a linear unconstrained MPC controller derived from the same cost function.

The structure of the paper is as follows. In section 2, we introduce the DSS framework proposed by [22] and the control objective of DSS; then using this framework, a modified DSS framework with a strict separation of the physical and numerical components is developed. Based on the modified framework, a reduced-order DSS observer is designed and the MPC controller formulation issue is discussed in section 3. In section 4, a QM system is studied: we first convert the system into a two-input, two-output DSS framework in the strict separation form; then, real-time application results show the applicability of the modified framework and the observer/MPC controller designs based upon it. Section 5 concludes the paper with a summary of the main achievements.

2 A DSS framework with strict separation of substructures

2.1 A brief introduction to the concise DSS framework [22,24] and its control objective

A general and concise DSS framework, as shown in the dash-dot box of Fig. 1, was originally proposed by [22] for SISO systems and then extended to MIMO cases [23, 24]. Based on this framework, a DSS system can be expressed by

$$y_N = G_1 d - G_0 u \tag{1}$$

$$y_P = G_2 u \tag{2}$$

Here, we can assume the transfer function G_1 represents the dynamics of the numerical substructure, G_2 the physical substructure and G_0 the interaction dynamics between the two substructures. In many cases this assumption holds, so that we can use the generalized set $\{\Sigma_N, \Sigma_P\}$ to represent the numerical and physical substructures, respectively, as shown in Fig. 1, where $\{y_N, y_P\}$ are the interface responses from $\{\Sigma_N, \Sigma_P\}$ and f is the interface constraint signal. The control objective is to use a control signal u to reduce the magnitude of the DSS substructuring error $y := y_N - y_P$, subject to the interface constraint f, when an external testing signal d is applied. The DSS control system design based on this framework usually contains a two degree of freedom controller: one feed-forward controller K_d, used to shape the testing signal, and a

feedback controller K_y, used to reduce the influence of the uncertainties and measurement noise. Since the testing signal can be assumed to be a measured disturbance, the DSS control can be viewed as an output regulation problem with measured disturbance rejection. The controller designs in [9, 22–24] are all based on this two degree of freedom DSS control framework.

2.2 A modified DSS framework with strict separation of physical and numerical components

In the original DSS framework, introduced in the last subsection, it is noted that the transfer system G_0 comprises both numerical and physical components. In this paper, the MPC design is based on a modified DSS framework that completely separates the numerical and physical components in the transfer system. This is motivated by the following: 1) this modified framework represents engineering aspects of DSS in a more reasonable way, where the physical and numerical substructures are usually required to be strictly separated; 2) it is convenient to perform system identification on the physical components only; 3) when designing an observer, the states of the numerical substructure can be viewed as noise-free outputs, which would lead to a reduced-order Kalman-Bucy filter; 4) it is convenient for performance analysis and robust controller design by representing uncertainties in the physical substructure only. The last point is beyond the scope of this paper and will be investigated in future work. Hence, we propose a modified DSS framework as illustrated in Fig. 2, on which the observer and MPC controller designs will be based.

In this modified DSS framework, the physical and numerical substructures are

$$G_P = \begin{bmatrix} G_{P0}G_{act} \\ G_{act} \end{bmatrix} \qquad\qquad G_N = \begin{bmatrix} G_{N1} & -G_{N0} \end{bmatrix}$$

Note that this modified framework is equivalent to the original framework via the relationships:

$$G_0 = G_{N0}G_{P0}G_{act} \qquad G_1 = G_{N1} \qquad G_2 = G_{act} \tag{3}$$

In the following, we represent G_P and G_N by their discrete time minimal realizations:

$$G_P \sim \begin{cases} x_P(k+1) &= A_P x_P(k) + B_P u(k) \\ y_I(k) &= C_{PI} x_P(k) + D_{PI} u(k) + v_I(k) \\ y_P(k) &= C_{PP} x_P(k) + v_P(k) \end{cases} \tag{4}$$

$$G_N \sim \begin{cases} x_N(k+1) &= A_N x_N(k) + B_{Nd} d(k) + B_{NI} y_I(k) \\ y_N(k) &= C_N x_N(k) \end{cases} \tag{5}$$

with $u(k), y_P(k), y_N(k) \in \mathbb{R}^m$, $d(k) \in \mathbb{R}^{n_d}$, $x_N(k) \in \mathbb{R}^{n_n}$ and $x_P(k) \in \mathbb{R}^{n_p}$. Here $v_I(k)$ and $v_P(k)$ are the interface measurement noise and the output measurement noise of the physical substructure respectively, and $y_I(k)$ is the constraint variable between the numerical and physical substructures. Note that we assume G_N is strictly proper and there is no direct feed-through term from the input to output in G_P. This assumption simplifies the following development without loss of generality; the case with non-strictly proper G_P and G_N can be determined in a straightforward manner.

Augmenting G_N and G_P, the state space realization for the whole DSS system is given by:

$$\begin{cases} x(k+1) &= Ax(k) + B_d d(k) + B_u u(k) + B_v v_I(k) \\ y(k) &= Cx(k) - v_P(k) \end{cases} \tag{6}$$

where $x(k) = [x_N^T(k), x_P^T(k)]^T \in \mathbb{R}^n$ with $n = n_n + n_p$ and

$$A = \begin{bmatrix} A_N & B_{NI}C_{PI} \\ 0 & A_P \end{bmatrix} \qquad B_d = \begin{bmatrix} B_{Nd} \\ 0 \end{bmatrix} \qquad B_u = \begin{bmatrix} B_{NI}D_{PI} \\ B_P \end{bmatrix}$$

$$C = \begin{bmatrix} C_N & -C_{PP} \end{bmatrix} \qquad B_v = \begin{bmatrix} B_{NI} \\ 0 \end{bmatrix}$$

Note that the DSS model (6) is strictly proper, which results from the fact that both the actuator model G_{act} and the numerical substructure model G_N are themselves strictly proper.

3 DSS observer design and MPC controller formulation

3.1 DSS observer design

Based on the modified DSS framework, it is convenient to design a reduced-order observer, which can result in less computational burden than a full-order observer designed by (6). This is especially beneficial when the on-line computation time is large. From the modified DSS framework, we can see that: 1) the numerical substructure model is exactly known, so that its states can be derived directly; 2) the measurement noise only contaminates the outputs of the physical substructure, while the states of the numerical substructure are completely noise-free. These two features lead naturally to the classical reduced-order Kalman-Bucy observer design problem.

We reformulate the DSS system (6) for the observer design as follows:

$$\begin{cases} x(k+1) &= Ax(k) + B_d d(k) + B_u u(k) + B_v v_I(k) \\ y_1(k) &= y(k) = Cx(k) - v_P(k) \\ y_2(k) &= x_N(k) \end{cases} \tag{7}$$

The output of the DSS system expressed by (7) contains both noisy and noise-free observed variables $y_1(k)$ and $y_2(k)$, which would lead to a singularity when resolving the discrete algebraic Riccati equation (DARE). Standard results are available in the literature for this problem, see e.g., [2] for the continuous time case and [25] for the discrete time case. Following a similar procedure as in [2, 25] we can design a reduced order DSS observer as follows.

From (4) and (5), we form the observation problem from the equations:

$$x_p(k+1) = A_P x_P(k) + B_P u(k) \tag{8}$$

$$\begin{bmatrix} y(k) \\ x_N(k+1) \end{bmatrix} = \tilde{C} x_P(k) + \tilde{D}_u u(k) + \tilde{D}_d d(k) + \tilde{D}_{x_N} x_N(k) + \tilde{D}_v v(k) \tag{9}$$

with

$$\tilde{C} = \begin{bmatrix} -C_{PP} \\ B_{NI} C_{PI} \end{bmatrix} \qquad \tilde{D}_u = \begin{bmatrix} 0 \\ B_{NI} D_{PI} \end{bmatrix} \qquad \tilde{D}_d = \begin{bmatrix} 0 \\ B_{Nd} \end{bmatrix}$$

$$\tilde{D}_{x_N} = \begin{bmatrix} C_N \\ A_N \end{bmatrix} \qquad \tilde{D}_v = \begin{bmatrix} 0 & -I \\ B_{NI} & 0 \end{bmatrix} \qquad v(k) = \begin{bmatrix} v_I(k) \\ v_P(k) \end{bmatrix}$$

From (8) and (9), we can see that there is no state excitation noise and if $V = E(v(k)v(k)^T) > 0$, then this observation problem is nonsingular. The observer gain is $K = A_P^T X \tilde{C}^T (\tilde{C}^T X \tilde{C} + V)^{-1}$, where X is the solution of the following DARE:

$$A_P X A_P^T + A_P X \tilde{C}^T (\tilde{C} X \tilde{C}^T + V)^{-1} \tilde{C} X A_P^T - X = 0 \tag{10}$$

Defining $z(k) := \hat{x}_p(k) - K_2 x_N(k)$ with $\hat{x}_P(k)$ as the estimated value of $x_P(k)$, the observer dynamics are given as

$$z(k+1) = A_o z(k) + L_{x_N} x_N(k) + L_y y(k) + L_u u(k) + L_d d(k) \tag{11a}$$

$$\hat{x}(k) = Q_1 x_N(k) + Q_2 (z(k) + K_2 x_N(k)) \tag{11b}$$

where

$$A_o = A_P - K\tilde{C} \qquad L_{x_N} = (A_P - K\tilde{C})K_2 - K\tilde{D}_{x_N} \qquad Q_1 = \begin{bmatrix} I_{n_n} \\ 0 \end{bmatrix} \qquad Q_2 = \begin{bmatrix} 0 \\ I_{n_p} \end{bmatrix}$$

$$L_u = B_P - K\tilde{D}_u \qquad L_d = -K\tilde{D}_d \qquad L_y = K_1$$

where K is partitioned as $K = [K_1, K_2]$ in accordance with the vector $[y(k)^T, x_N(k+1)^T]^T$. Here the observer output is $\hat{x}(k) := [x_N(k)^T, \hat{x}_P(k)^T]^T$, which is comprised of the estimated states of the physical substructure and the states of the numerical substructure. Compared with the observer designed directly by (6), this observer has n_p states, which leads to less computational burden. Furthermore, if the measurement noise on some of the DSS physical substructure outputs can be neglected, the DSS observer order can be further reduced.

3.2 MPC controller development

The manipulated control input provided by the MPC controller is determined by

$$u(k) = \bar{E}U^*(k)$$

where $\bar{E} = [I_m,\, 0, \ldots, 0] \in \mathbb{R}^{m \times H_p m}$ with H_p as the prediction horizon and $U^*(k)$ is the optimizer of the following optimization problem:

$$\begin{aligned} U^*(k) = \arg\min_{U(k)} & J(k) \\ & \text{subject to} \\ & x(k+1) = Ax(k) + B_d d(k) + B_u u(k) \\ & y(k) = Cx(k) \\ & u_{\min} \preceq \hat{u}(k+i|k) \preceq u_{\max} \text{ with } i = 0, 1, \ldots, H_p - 1 \\ & y_{\min} \preceq \hat{y}(k+i|k) \preceq y_{\max} \text{ with } i = 1, \ldots, H_p \end{aligned} \tag{12}$$

where $U(k) = \left[\hat{u}(k|k)^T, \hat{u}(k+1|k)^T, \cdots, \hat{u}(k+H_p-1|k)^T\right]^T$ and "$\preceq$" denotes componentwise inequality. $J(k)$ is a cost function, which can take various forms. Suppose this cost function has the following quadratic form:

$$J(k) = \|\hat{y}(k+H_p|k)\|_P^2 + \sum_{i=1}^{H_p-1} \|\hat{y}(k+i|k)\|_Q^2 + \sum_{i=0}^{H_p-1} \|\hat{u}(k+i|k)\|_R^2 \tag{13}$$

where Q, R and P are the weights for the output, input, and the terminal output $\hat{y}(k+H_p|k)$, respectively. The terminal weight P can be calculated by a solution of a discrete Lyapunov function or a DARE [11]:

$$P = A^T[P - PB(B^TPB + R)^{-1}B^TP]A + Q$$

which is used to account for the cost function beyond the H_p horizon. The inclusion of P can improve the performance the MPC controller (with a fixed-length horizon) to some extent and the stability can be guaranteed when H_p is sufficiently large [3, 12, 18].

To resolve the optimization problem, the cost function needs to be converted into a quadratic programme (QP) first of all. By iterating the plant dynamic equations (6) with the measurement noise $v_p = 0$ and $v_I = 0$, we have

$$Y(k) = \Phi_x x(k) + \Phi_U U(k) + \Phi_d d(k) \tag{14}$$

with

$$\Phi_x = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{H_p} \end{bmatrix} \quad \Phi_U = \begin{bmatrix} CB_u & & & \\ CAB_u & CB_u & & \\ \vdots & \vdots & \ddots & \\ CA^{H_p-1}B_u & CA^{H_p-2}B_u & \cdots & CB_u \end{bmatrix} \quad \Phi_d = \begin{bmatrix} CB_d \\ CAB_d + CB_d \\ \vdots \\ \sum_{i=0}^{H_p-1} CA^i B_d \end{bmatrix}$$

Here, the disturbance $d(k)$ is measured at the same time as the measurement of $y(k)$. The future estimation of $\hat{d}(k+i|k)$ is influenced by the knowledge of the behaviour of the disturbance. In (14), it is assumed that the estimation of $\hat{d}(k+i|k)$ is constant, i.e. $d(k) = \hat{d}(k+1|k) = \ldots = \hat{d}(k+H_p-1|k)$, [11].

Substituting (14) into (13) and (12) leads to a quadratic programme (QP)

$$\begin{aligned} U^*(k) = &\arg\min_{U(k)} \left[\frac{1}{2} U(k)^T H U(k) + U(k)^T (F_x \hat{x}(k) + F_d d(k)) \right] \\ &\text{subject to } LU(k) \preceq b(k) \end{aligned} \tag{15}$$

where

$$H = \Phi^T \mathcal{Q} \Phi + \mathcal{R} \qquad F_x = \Phi^T \mathcal{Q} \Lambda \qquad F_d = \Phi^T \mathcal{Q} \Phi_d$$

and

$$\mathcal{Q} = \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & P \end{bmatrix} \qquad \mathcal{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}$$

$$L = \begin{bmatrix} I_{nu \times Hp} \\ -I_{nu \times Hp} \\ \Phi_U \\ -\Phi_U \end{bmatrix} \qquad b(k) = \begin{bmatrix} U_{\max} \\ -U_{\min} \\ Y_{\max} - \Phi_x x(k) - \Phi_d d(k) \\ -Y_{\min} + \Phi_x x(k) + \Phi_d d(k) \end{bmatrix}$$

with $U_{\max} = \left[u_{\max}^T, \ldots, u_{\max}^T\right]^T$, $U_{\min} = \left[u_{\min}^T, \ldots, u_{\min}^T\right]^T$, $Y_{\max} = \left[y_{\max}^T, \ldots, y_{\max}^T\right]^T$ and $Y_{\min} = \left[y_{\min}^T, \ldots, y_{\min}^T\right]^T$.

From equation (15), we can see that when the input constraints are ignored, a linear controller can be determined by:

$$u_k = \bar{E}U(k)^* = -(K_x\hat{x}(k) + K_d d(k)) \tag{16}$$

with $K_x = \bar{E}H^{-1}\Phi_x$ and $K_d = \bar{E}H^{-1}\Phi_d$. Note that (16) contains a feedforward term K_d and a feedback term K_x. In this paper we denote the controller (16) as a linear unconstrained MPC and we note that the MPC controller formulated from (15) also has two-degrees of freedom. However, the main difference between the MPC controller and the linear unconstrained MPC controller (16) is that the MPC controller can achieve an optimal solution subject to the constraints. In the experimental implementation described later in this paper, a performance comparison is made between the MPC controller and the linear unconstrained controller (16).

We now summarize the main steps and issues that need to be considered when designing an MPC controller for DSS:

1) DSS framework establishment. Formulate a strict separation DSS framework and derive its corresponding state space realization according to (6). It is sometimes essential to identify the models of the physical substructure components, including the actuators, using system identification methods.

2) DSS observer design. If the measurement noise from the physical substructure is significant, a Kalman-Bucy observer with a reduced order of n_p can be designed according to (11); when some of the measurement noise can be neglected, a DSS observer with a further reduced order can be designed.

3) MPC controller formulations. When the actuator slew rate limit needs to be considered, a more generic cost function should be employed:

$$\begin{aligned} J(k) &= \|\hat{y}(k+H_p|k)\|_P^2 \\ &+ \sum_{i=0}^{H_p-1}\left[\|\hat{y}(k+i|k)\|_Q^2 + \|\hat{u}(k+i|k)\|_R^2\right] + \sum_{i=0}^{H_u-1}\|\Delta\hat{u}(k+i|k)\|_S^2 \end{aligned} \tag{17}$$

with $\Delta\hat{u}(k+i|k) = \hat{u}(k+i|k) - \hat{u}(k+i-1|k)$. This cost function can also be formulated into a QP by augmenting the original system with the state $[x(k)^T, u(k-1)^T]^T$ and the matrices

$$\tilde{A} = \begin{bmatrix} A & B_u \\ 0 & I \end{bmatrix} \qquad \tilde{B}_u = \begin{bmatrix} B_u \\ I \end{bmatrix} \qquad \tilde{B}_d = \begin{bmatrix} B_d \\ 0 \end{bmatrix} \qquad \tilde{C} = [C \quad 0] \tag{18}$$

The terminal weight can be calculated from a DARE solution of $\tilde{P}$, satisfying:

$$\tilde{P} = \tilde{A}^T \tilde{P} \tilde{A} + \tilde{Q} - (\tilde{A}^T \tilde{P} \tilde{B} + \tilde{N})^T (\tilde{B}^T \tilde{P} \tilde{B} + \tilde{R})^{-1} (\tilde{B}^T \tilde{P} \tilde{A} + \tilde{N}^T) \tag{19}$$

with

$$\tilde{Q} = \begin{bmatrix} C^T Q C & 0 \\ 0 & R \end{bmatrix} \qquad \tilde{N} = \begin{bmatrix} 0 \\ R \end{bmatrix} \qquad \tilde{R} = R + S \tag{20}$$

Based on this augmentation, the MPC controller can be formulated in a similar way to (15). The corresponding unconstrained MPC also takes the same form as (16).

It should be noted that various MPC formulations exist for different problems. For example, when A is ill-conditioned, a linear transformation $u(k) = Kx(k) + r(k)$ can be introduced [20], [17], so that the resulting MPC controller formulation is based on the system $x(k+1) = (A + B_u K)x(k) + B_u r(k) + B_d d(k)$.

4) Feasibility and stability of MPC. Guaranteeing the feasibility and stability of MPC is not an easy theoretical problem to solve, especially when the output limits are also considered. When the prediction horizon is sufficiently large, the inclusion of a terminal weight, terminal cost function and a local controller can guarantee feasibility and stability [12]. However, this is not easy to check numerically. In applications, it is common practice to soften the output constraints to make the QP feasible [11]. Note that for the cost function (17), the horizon for $\Delta\hat{u}(k+i|k)$ can be shorter than H_p, while the stability is still guaranteed (see, e.g. [19]). Furthermore, it could be useful to perform a stability and robustness test of the MPC system, for example, using the methods developed in [5, 8, 10], especially when the testing of a DSS is costly in a civil engineering application, for example.

5) MPC controller tuning. The tuning parameters mainly include the prediction horizon H_p (and a shorter input changing rate horizon H_u for (17)), the weighting matrices Q and R, and the constraints. Furthermore, a trade-off is required between the maximum allowable sampling time, the saturation limit and the tuning parameters, in order to achieve the best performance. When the actuator limits are relatively large, the output weight can be increased and the output limits can be more restrictive.

4 Case study – a quasi motorcycle (QM) system

We consider a QM system, which has been developed at the University of Bristol; see Fig 5. The testing rig is comprised of three subsystems, the first of which is a rigid vehicle body with an evenly distributed mass of 229kg and two suspension struts. These are connected to the vehicle body and the other ends are connected via swing arms to two 25kN hydraulic actuators. The second and third subsystems consist of two further 25kN hydraulic actuators, attached to the hubs of the front and rear wheels/tyres. Each hydraulic actuator has a built-in linear variable displacement transformer (LVDT) for the measurement of displacement and also a load cell for the measurement of force. Two extra LVDTs are used to measure the extensions of the suspension struts. In this case study, we select the subsystems as follows: the QM body with the two suspension struts is the physical substructure, while the front and rear wheels form the numerical substructure. This arrangement is shown schematically in Fig. 6, where the swing arms are omitted for simplicity, by scaling the stiffness and damping parameters of the suspension struts appropriately. We call this a *single mode* substructure since only one type of force, i.e. inertial terms, occur in the physical substructure. Alternatively, we can also model one wheel numerically and the other physically, or two wheels physically and the body with two suspension struts numerically, depending on the problems that we are interested in (see [23, 24] for a detailed discussion). The control objective is to synchronize the physical and numerical substructures by minimizing the error between the measured displacements of the bases of the front/rear suspension struts and the numerically generated displacements of the front/rear wheel hubs, subject to the interaction force constraint between the bases of the suspension struts and the wheel hubs. In the following, we first convert this single mode QM suspension system into the modified DSS framework. Then, real-time application results are presented to show the performance of the MPC controller when synchronizing this substructured system. The QP in the MPC controller is solved using the routines in [28], which can guarantee a reasonable on-line computing speed [29]. The notation for the variables and parameters, as well as the values of parameters, are listed in the Appendix.

4.1 Model establishment for the QM suspension system and its representation in the modified DSS framework

The dynamic equations of the quasi-motorcycle body are:

$$m_3\ddot{y}_3 = f_1 + f_2 - m_3 g \tag{21}$$

$$J\ddot{\theta} = (L_2 f_2 - L_1 f_1)\cos\theta \tag{22}$$

with $\sin\theta = (y_{31} - y_{32})/L$ and $y_3 = (L_2 y_{31} + L_1 y_{32})/L$. Here the approximations $\sin\theta \approx \theta$ and $\cos\theta \approx 1$, for a small value of θ, are assumed.

The dynamic equations for the front and rear wheels are:

$$\begin{aligned} m_1\ddot{y}_{w1} =& k_1(d_1 - y_{w1}) + c_1(\dot{d}_1 - \dot{y}_{w1}) - f_1 - m_1 g \\ m_2\ddot{y}_{w2} =& k_2(d_2 - y_{w2}) + c_2(\dot{d}_2 - \dot{y}_{w2}) - f_2 - m_2 g \end{aligned}$$

The dynamic equations for the front and rear ends of the QM body are:

$$\begin{aligned} \frac{L_2}{L} m_3\ddot{y}_{31} &= k_{31}(y_{a31} - y_{31}) + c_{31}(\dot{y}_{a31} - \dot{y}_{31}) - \frac{L_2}{L} m_3 g \\ \frac{L_1}{L} m_3\ddot{y}_{32} &= k_{32}(y_{a32} - y_{32}) + c_{32}(\dot{y}_{a32} - \dot{y}_{32}) - \frac{L_1}{L} m_3 g \end{aligned}$$

and, based on experimental system identification, the dynamics of the inner-loop controlled actuators are represented by second order linear models:

$$y_{a31}(s) = \underbrace{\left(\frac{7298}{s^2 + 170.8s + 6876}\right)}_{G_{act1}} u_1(s) \qquad y_{a32}(s) = \underbrace{\left(\frac{7690}{s^2 + 162.2s + 7603}\right)}_{G_{act2}} u_2(s)$$

Now we choose the forces produced by the suspension struts f_1 and f_2 as the interaction constraints between the two substructures, the displacements wheel hubs y_{w1} and y_{w2} as the outputs of the numerical substructure and the displacements of the actuators y_{a31} and y_{a32} as the outputs of the physical substructure. Hence the dynamics of the DSS system can be represented as:

$$y_N = G_{N1} d - G_{N0} G_{P0} G_{act} u \tag{23}$$

$$y_P = G_{act} u \tag{24}$$

where

$$y_N = \begin{bmatrix} y_{w1} \\ y_{w2} \end{bmatrix} \qquad y_P = \begin{bmatrix} y_{a31} \\ y_{a32} \end{bmatrix} \qquad d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{25}$$

$$G_{N1} = \begin{bmatrix} G_{yd1} & 0 \\ 0 & G_{yd2} \end{bmatrix} \qquad G_{N0} = \begin{bmatrix} G_{yf1} & 0 \\ 0 & G_{yf2} \end{bmatrix} \tag{26}$$

$$G_{P0} = \begin{bmatrix} P_2 s^2 \cdot G_{P01} & P_3 s^2 \cdot G_{P02} \\ P_3 s^2 \cdot G_{P01} & P_1 s^2 \cdot G_{P02} \end{bmatrix} \qquad G_{act} = \begin{bmatrix} G_{act1} & 0 \\ 0 & G_{act2} \end{bmatrix} \tag{27}$$

with

$$G_{yd1} = \frac{c_1 s + k_1}{m_1 s^2 + c_1 s + k_1} \qquad G_{yd2} = \frac{c_2 s + k_2}{m_2 s^2 + c_2 s + k_2}$$
$$G_{yf1} = \frac{1}{m_1 s^2 + c_1 s + k_1} \qquad G_{yf2} = \frac{1}{m_2 s^2 + c_2 s + k_2}$$
$$G_{P01} = \frac{(c_{31} s + k_{31})}{\frac{L_2}{L_3} m_3 s^2 + c_{31} s + k_{31}} \qquad G_{P02} = \frac{(c_{32} s + k_{32})}{\frac{L_1}{L_3} m_3 s^2 + c_{32} s + k_{32}}$$

and

$$P_1 = m_3 L_1^2 / L^2 + J/L^2$$
$$P_2 = m_3 L_2^2 / L^2 + J/L^2$$
$$P_3 = m_3 L_1 L_2 / L^2 - J/L^2$$

The control objective is to minimize the DSS substructuring error $y = y_N - y_P$.

Here the physical and numerical substructures are:

$$G_P = \begin{bmatrix} G_{P0} G_{act} \\ G_{act} \end{bmatrix} \qquad G_N = \begin{bmatrix} G_{N1} & -G_{N0} \end{bmatrix}$$

Note that G_{P0} is not proper but the term $G_{P0} G_{act}$ is proper; hence the physical part is proper. G_{N1} and G_{N0} share some common modes, hence a minimal realization of G_N should be used. The resulting QM model represented in DSS framework has the following dimensions: $x_N \in \mathbb{R}^4$, $x_P \in \mathbb{R}^8$, $x \in \mathbb{R}^{12}$, $y \in \mathbb{R}^2$ and $u \in \mathbb{R}^2$.

4.2 Experimental results

A dSPACE RS1103 system with Control Desk (Release 6.0) was used to implement the real-time control. The QP used for the MPC controller was based on the active set algorithm. We used the routine programmed in C language by [28], and developed a level-2 S function under MATLAB 2007b, which was compiled by the dSPACE compiler (rti1103). The SIMULINK block diagram of the MPC controller based on the cost function (17) and the augmentation (18), (19) and (20)

is shown in Fig.4, where T_D and T_I are $n \times n$ diagonal matrices with elements t_D and t_I as given by

$$t_D = \frac{1}{z} \qquad\qquad t_I = \frac{z}{z-1}$$

It is noted that the formulation of the slew rate and the input constraint term $b(k)$ accounts for part of the on-line computation. The MPC controller block diagram associated with the cost function (13) takes on a simpler form than shown in Fig. 4, hence it is ignored here.

The testing signal $d = [d_1, d_2]$ was composed of two ramp chirp signals, where d_2 had a 0.85s delay from d_1. The ramping time was 20s with the magnitude increasing from 0m to 0.0025m and the frequency span was from 15Hz to 2Hz. This testing signal was assumed to be a road disturbance, when the vehicle (1.7m in length between the front and rear wheels) was running at a speed of 2m/s.

The MPC controller shown in Fig. 4 and its corresponding unconstrained MPC were implemented on the QM system and a reduced order observer based on (11) was synthesized. A process noise w was assumed to be added to the equation (8) for tuning purposes and the ratio of the weights on the process noise and the measurement noise was chosen as 10:1. We enforced a hard limit of ± 0.002m on the displacement magnitude and a hard limit of ± 0.1m/s on the slew rate of both actuators. For the unconstrained MPC, the control signal was reset to the upper or lower limit values, when its calculated value exceeded its limits. The hard input limits used here are purely for demonstration purposes; the real input limits of the hydraulic actuators are higher than these values.

The weights on Q, R and S in the cost function (17) significantly affect the robust stability and the performance of the system: decreasing Q can make the system more stable and produces a smaller input energy, but the DSS error will be larger; on the other hand, increasing Q can give the converse effects. By trial and error, we chose the weights as $Q = \text{diag}(5, 5)$, $R = \text{diag}(0.1, 0.1)$ and $S = \text{diag}(1, 1)$. This choice of weights led to a bandwidth of 16Hz of the closed loop unconstrained MPC system. We used a safe sampling rate of 500Hz, which is over 30 times the closed-loop bandwidth.

Since the length of the prediction horizon influences the computation burden to a great extent, a maximum value should be chosen. By trying different values on H_p and H_u in experiments, we determined suitable prediction horizons as $H_p = 5$ and $H_u = 3$. This allowed the sampling of 500Hz, while further increasing the horizons did not improve the performance significantly.

One set of experimental results are shown in Fig. 7, based on the above choice of parameters. From the Figs. 7(a) and 7(b), we can see that the magnitude of the substructuring errors resulting from the unconstrained MPC controller exceed ±0.002m, but the substructuring errors from the MPC controller are within ±0.001m. Figs 7(c) and 7(d) show that the input magnitude is within ±0.002m and Figs. 7(e) and 7(f) show that the input changing rate is within $\pm 2 \times 10^{-4}$m at a sampling frequency of 500Hz, which corresponds to a maximum slew rate of ±0.1m/s. Although the comparison of the inputs and input slew rates shown in the Figs. 7(c)−7(f) are not easy to discern, the accumulated sum of squares of the values $y(k)$, $u(k)$ and $\Delta u(k)$ plotted in Figs. 8(a) - 8(c) confirm the improved response from the MPC controller, compared with those from the unconstrained MPC controller.

In addition to the experiment described above, we have also conducted other tests, including the MPC controller associated with the cost function (13), with various choices of parameters and with observer of different orders. From these experiments, we have the following remarks. The order of observer and different formulations of MPC controller can influence the implementation time. When the MPC controller associated with the cost function (13), with a prediction horizon $H_p = 5$, was employed, a reduced-order observer with 4 states (in this case, the DSS outputs were assumed to be noise-free) allowed a sampling rate up to 1.2 kHz. A reduced-order observer with 6 states allowed a sampling rate of 1kHz and a full order observer with 12 states allowed a sampling rate of 700 Hz. The use of reduced-order observers was less demanding of computation time, while a similar performance to the full-order observer was still achieved. Furthermore, the implementation of the MPC controller associated with the cost function (17) needed more computation time, since the computation of the constraint was performed on-line.

5 Conclusion

We have developed a procedure for using an MPC strategy to synchronize multivariable DSS systems, subject to actuator magnitude saturation and actuator slew rate limits. A modified DSS framework with a strict separation of DSS components was proposed to facilitate the DSS observer design and the MPC controller formulation. A QM suspension system was demonstrated as an implementation case study. Real-time experiments on the test rig demonstrated the feasibility of applying the traditional on-line MPC strategy to DSS, based on the modified DSS framework.

6 Acknowledgement

The authors gratefully acknowledge the support of the UK Engineering & Physical Sciences Research Council, grant number: EP/D0036917, "The adaptive control of generalised dynamically substructured systems", in the pursuance of this work.

References

[1] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

[2] A. E. Bryson and D. E. Johansen. Linear filtering for time-varing systems using measurements containing colored noise. In *IEEE Trans. on Automatic Control*, volume 10, pages 4–10, 1965.

[3] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29:121–129, 1996.

[4] A. Glattfelder and W. Schaufelberger. *Control Systems with Input and Output*. Springer, 1 edition, 2003.

[5] W. P. Heath and G. Li. Improved multipliers for input-constrained model predictive control. In *17th IFAC World Congress*, pages 15160–15165, Seoul, Korea, 2008.

[6] T. Hu and Z. Lin. *Control Systems with Actuator Saturation*. Springer, 2001.

[7] V. Kapila and K. M. Grigoriadis. *Actuator Saturation Control*. CRC Press, 2002.

[8] G. Li, W. P. Heath, and B. Lennox. Concise stability conditions for systems with static nonlinear feedback expressed by a quadratic program. *IET Control Theory & Applications*, 2:554–563, 2008.

[9] G. Li, G. Herrmann, D. P. Stoten, J. Tu, and M. C. Turner. A disturbance rejection anti-windup framework and its application to a substructured system. In *the 47th IEEE Conference on Decision and Control*, pages 3510–3515, Cancun, Mexico, 2008.

[10] C. Løvaas, M. M. Seron, and G. C. Goodwin. Robust output-feedback model predictive control for systems with unstructured uncertainty. *Automatica*, 44:1933–1943, 2008.

[11] J. M. Maciejowski. *Predictice control with constraints.* Addison-Wesley Publishing Company, UK, 2002.

[12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[13] T. Nakashima, H. Kato, and E. Takaoka. Development of real-time pseudo dynamic testing. *Earthquake Engineering and Structural Dynamics*, 21:79–92, 1992.

[14] S. A. Neild, D. P. Stoten, D. Drury, and D. J. Wagg. Control issues relating to real-time substructuring experiments using a shaking table. *Earthquake Engineering and Structural Dynamics*, 34:1171–1192, 2005.

[15] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In *Fifth International Conference on Chemical Process Control, CACHE, AIChE*, pages 232–256, 1997.

[16] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[17] J. B. Rawlings. Tutorial overview of model predictive control. In *IEEE Control Systems Magazine*, pages 38–52, 2001.

[18] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. In *IEEE Trans. on Automatic Control*, volume 38, pages 1512–1516, 1993.

[19] J. A. Rossiter. *Model-based predictive control: a practical approach.* CRC Press, USA, 2003.

[20] J. A. Rossiter, M. J. Rice, and B. Kouvaritakis. A robust state-space approach to stable predictive control strategies. In *Proceedings of the American Control Conference*, pages 1640–1641, Albuquerque, New Mexico, 1997.

[21] D. P. Stoten and H. Benchoubane. Robustness of a minimal controller synthesis algorithm. *Internationl Journal of Control*, 51:851–861, 1990.

[22] D. P. Stoten and R. A. Hyde. Adaptive control of dynamically substructured systems: the single-input single-output case. In *Proc. IMechE Part I: Systems and Control Engineering*, volume 220, pages 63–79, 2006.

[23] D. P. Stoten, J. Tu, and G. Li. Adaptive control of generalised dynamically substructured systems. In *17th IFAC congress*, pages 14090–14095, Seoul, Korea, 2008.

[24] D. P. Stoten, J. Tu, and G. Li. Synthesis and control of generalised dynamically substructured systems. In *Proc. IMechE Part I: Systems and Control Engineering*, volume 223, pages 371–392, 2009.

[25] E. Tse and M. Athans. Optimal minimal-order observer-estimators for discrete linear time-varying systems. In *IEEE Trans. on Automatic Control*, volume 15, pages 416–426, 1970.

[26] D. J. Wagg and D. P. Stoten. Substructuring of dynamical systems via the adaptive minimal control synthesis algorithm. *Earthquake Engineering Structural Dynamics*, 30:865–877, 2001.

[27] M. S. Williams and A. Blakeborough. Laboratory testing of sturctures under dynamic loads: an introductory review. *Phil. Trans. R. Soc. Lond. A,*, 359:1651–1669, 2001.

[28] A. G. Wills. QPC - Quadratic Programming in C. School of Elec. Eng. & Comp. Sci., University of Newcastle, Australia , 2008. The software package is avalable from `http://sigpromu.org/quadprog/`.

[29] A. G. Wills, D. Bates, A. Fleming, B. Ninness, and R. Moheimani. Model Predictive Control applied to constraint handling active noise and vibration control. In *IEEE Transactions on Control Systems Technology*, volume 16, pages 3–12, 2008.

[30] M. N. Zeilinger, C. N. Jones, and M. Morari. Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Optimization. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4718–4723, Cancun, Mexico, 2008.

A Notation list for the QM system

A.1 Parameters

Notation	Description	Values
	Quasi-motorcycle body:	
m_3	Mass	229kg
J	Moment of inertia	62.7kgm^2
L	Body length	1.60m
L_1, L_2	Lengths from front/rear	0.800m, 0.800m
	end to mass center	
	Front/rear suspension:	
k_{31}, k_{32}	Stiffness.	34.8kN/m, 39.5kN/m
c_{31}, c_{32}	Damping	717Ns/m, 970Ns/m
	Front/rear wheels:	
m_1, m_2	Mass	12.3kg, 15.7kg
k_1, k_2	Stiffness	384kN/m, 409kN/m
c_1, c_2	Damping	700Ns/m, 816Ns/m

A.2 Variables

Notation	Description
y_{w1}, y_{w2}	Front/rear wheel displacements.
y_b	Body center of mass displacement.
θ	Pitch of the body.
y_{31}, y_{32}	Front/rear ends of body displacements.
y_{a1}, y_{a2}	Front/rear suspension base displacements.
	(the outputs of the front/rear suspension actuators).
u_1, u_2	Inputs of the front/rear suspension actuators.
f_1, f_2	Interaction forces.
d_1, d_2	Disturbances on the front/rear wheels.

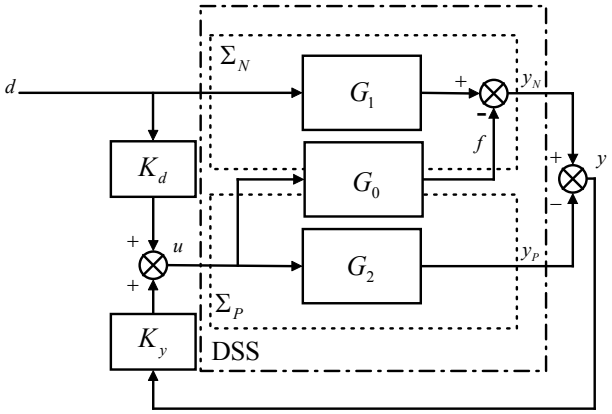


Figure 1: The DSS framework by [22] with DSS control

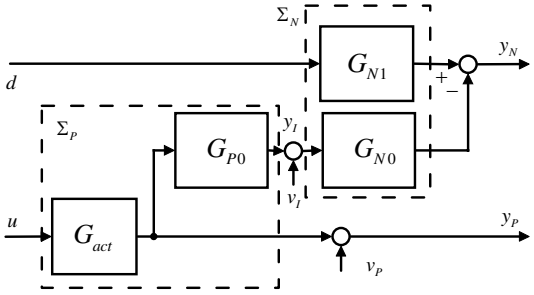


Figure 2: A modified DSS framework with a strict separation of physical and numerical components

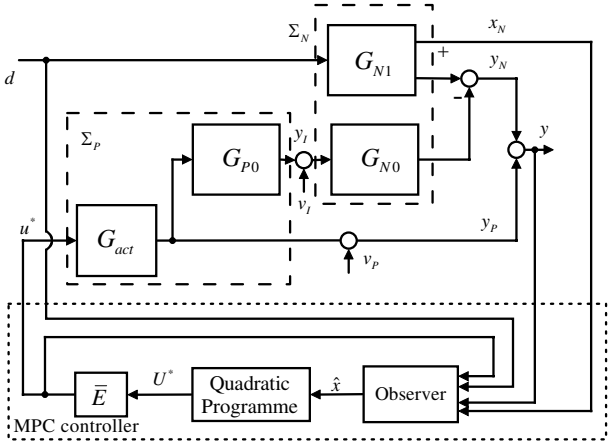


Figure 3: DSS framework with an MPC controller

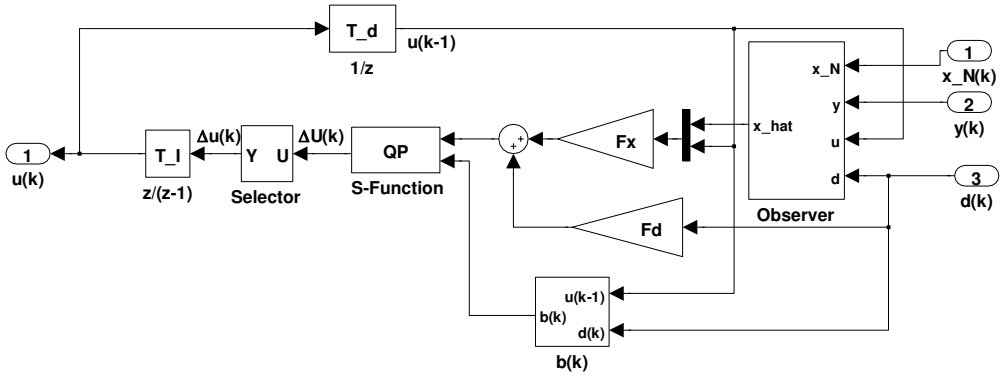


Figure 4: MPC controller formulation for the cost function (17)

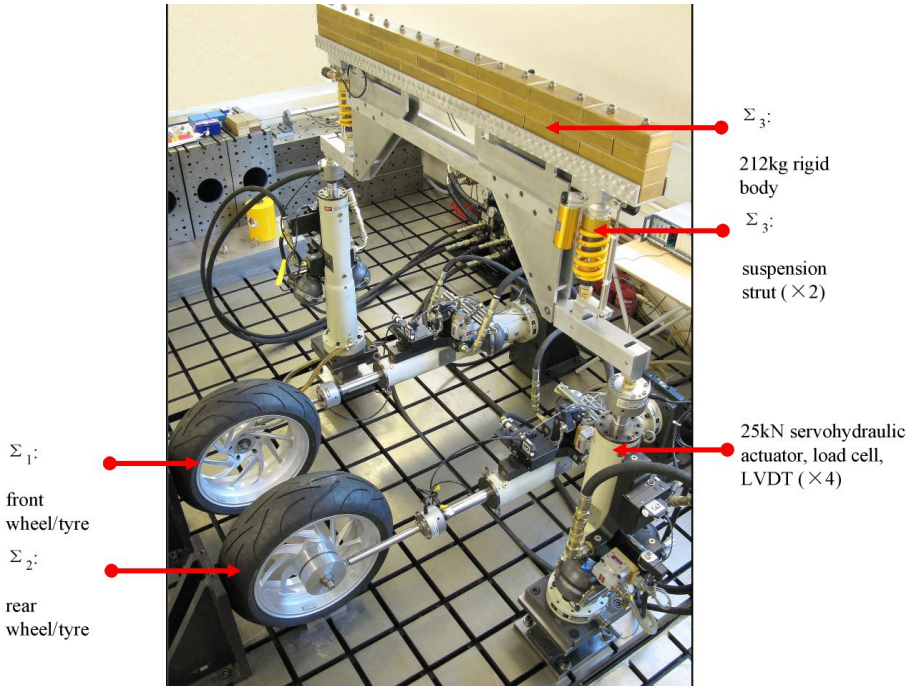


Figure 5: The quasi motorcycle system testing rig

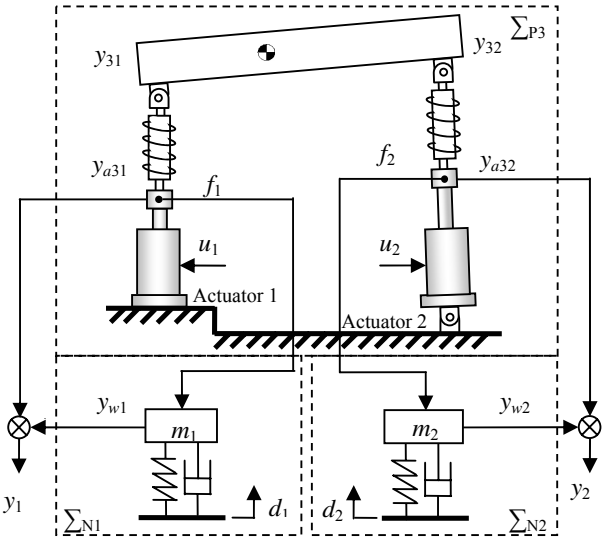
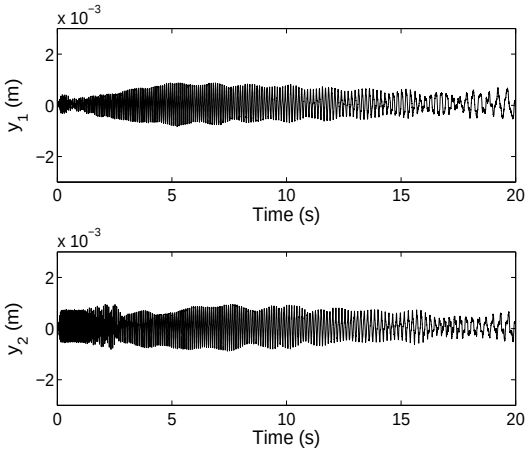
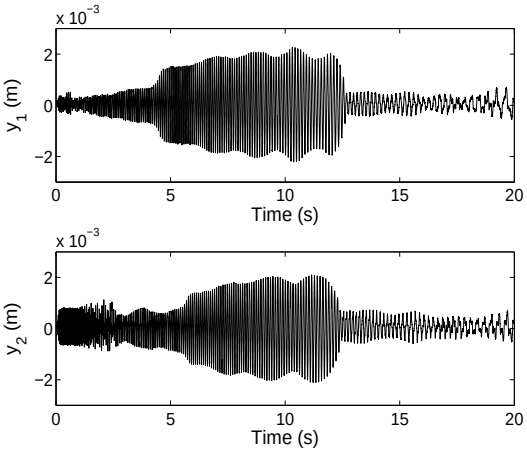


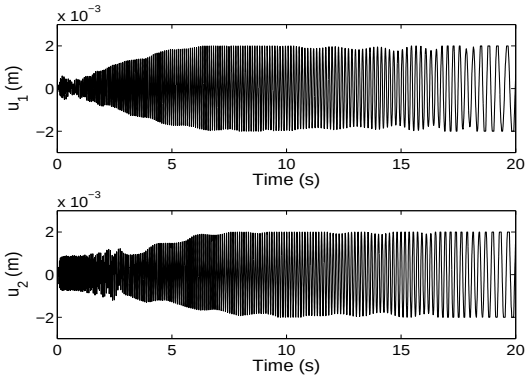
Figure 6: The schematic diagram for the QM system represented in the single mode DSS



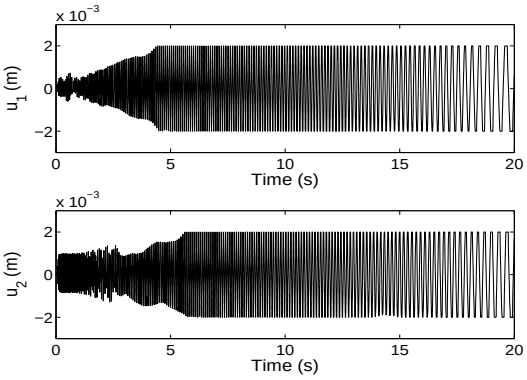
(a) The output y_1 and y_2 (MPC)



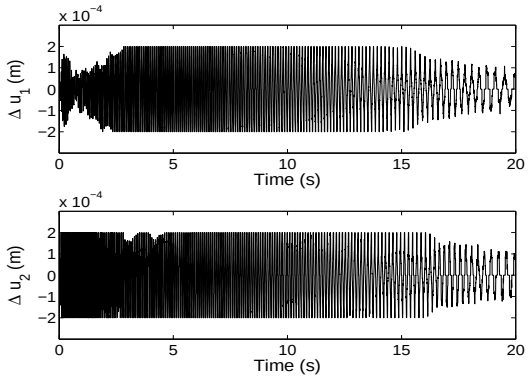
(b) The outputs y_1 and y_2 (unMPC)



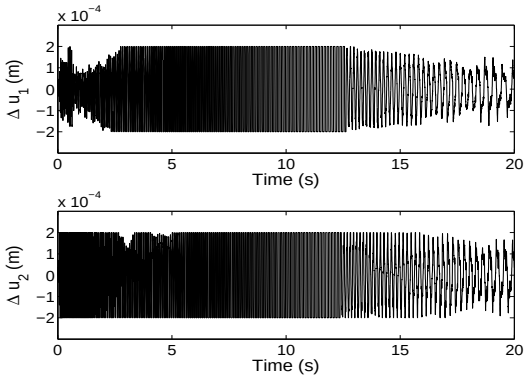
(c) The inputs u_1 and u_2 (MPC)



(d) The input of u_1 and u_2 (unMPC)

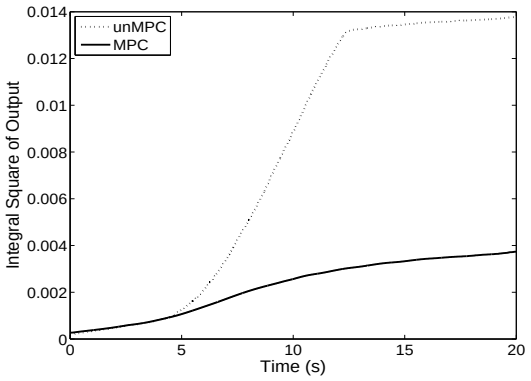


(e) The input slew rates Δu_1 and Δu_2 (MPC)

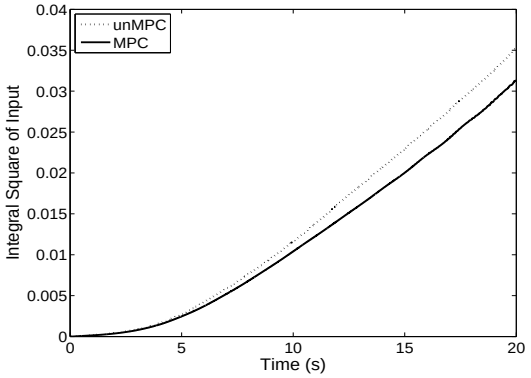


(f) The input slew rates Δu_1 and Δu_2 (unMPC)

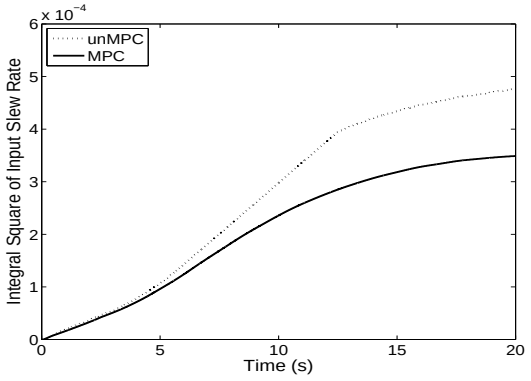
Figure 7: A comparison between the DSS for QM controlled by MPC and unconstrained MPC (unMPC) with trimmed input and trimmed input slew rate.



(a) The sum of squares of the output



(b) The sum of squares of the input



(c) The sum of squares of the input slew rate

Figure 8: A comparison of the sum of square of the DSS output, input and input slew rate between the MPC and unconstrained MPC with trimmed inputs.